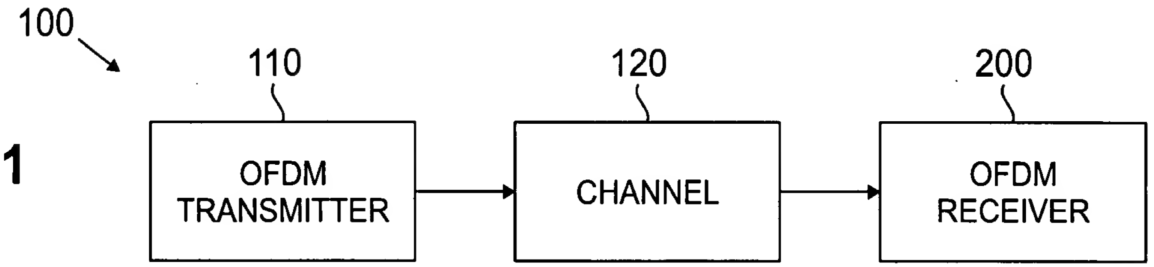

100
110
OFDM TRANSMITTER
120
CHANNEL
200
OFDM RECEIVER

FIG. 1

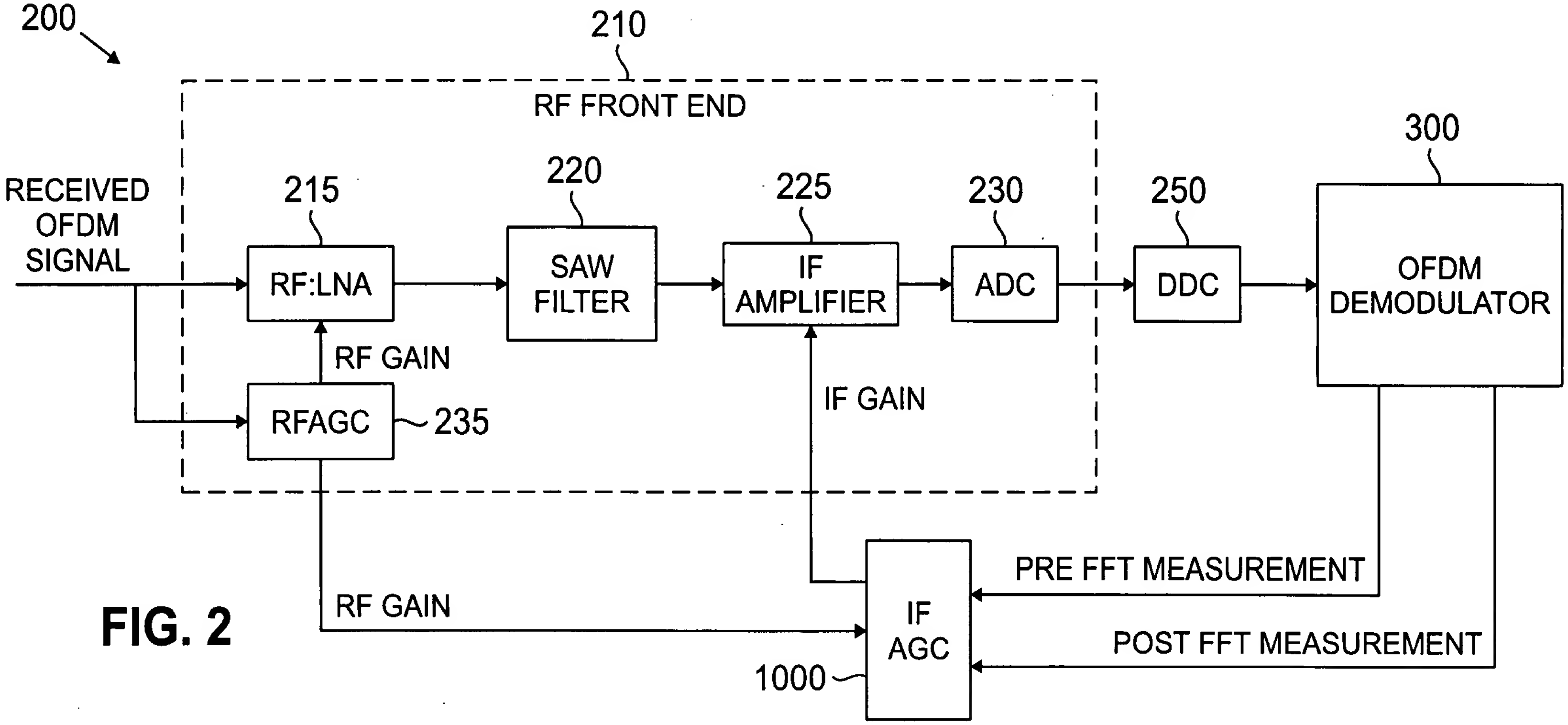

200
210
RF FRONT END
RECEIVED OFDM SIGNAL
215
RF:LNA
RF GAIN
RFAGC
235
220
SAW FILTER
225
IF AMPLIFIER
IF GAIN
230
ADC
250
DDC
300
OFDM DEMODULATOR
PRE FFT MEASUREMENT
POST FFT MEASUREMENT
RF GAIN
IF AGC
1000

FIG. 2

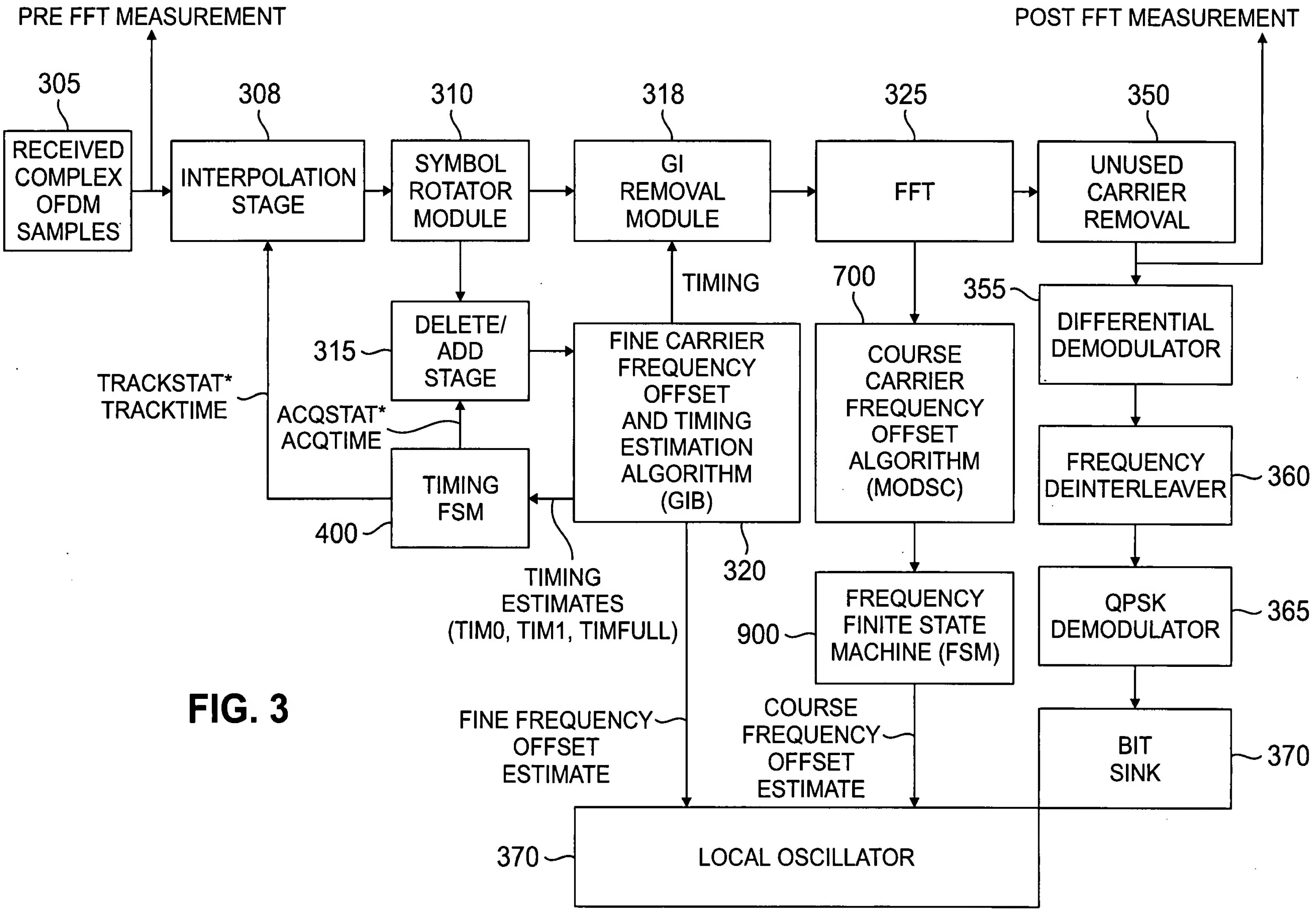

PRE FFT MEASUREMENT
POST FFT MEASUREMENT
305
RECEIVED COMPLEX OFDM SAMPLES
308
INTERPOLATION STAGE
310
SYMBOL ROTATOR MODULE
318
GI REMOVAL MODULE
325
FFT
350
UNUSED CARRIER REMOVAL
TIMING
700
355
DIFFERENTIAL DEMODULATOR
315
DELETE/ ADD STAGE
FINE CARRIER FREQUENCY OFFSET AND TIMING ESTIMATION ALGORITHM (GIB)
COURSE CARRIER FREQUENCY OFFSET ALGORITHM (MODSC)
TRACKSTAT* TRACKTIME
ACQSTAT* ACQTIME
TIMING FSM
FREQUENCY DEINTERLEAVER
360
400
320
TIMING ESTIMATES (TIM0, TIM1, TIMFULL)
900
FREQUENCY FINITE STATE MACHINE (FSM)
QPSK DEMODULATOR
365
FINE FREQUENCY OFFSET ESTIMATE
COURSE FREQUENCY OFFSET ESTIMATE
BIT SINK
370
370
LOCAL OSCILLATOR

FIG. 3

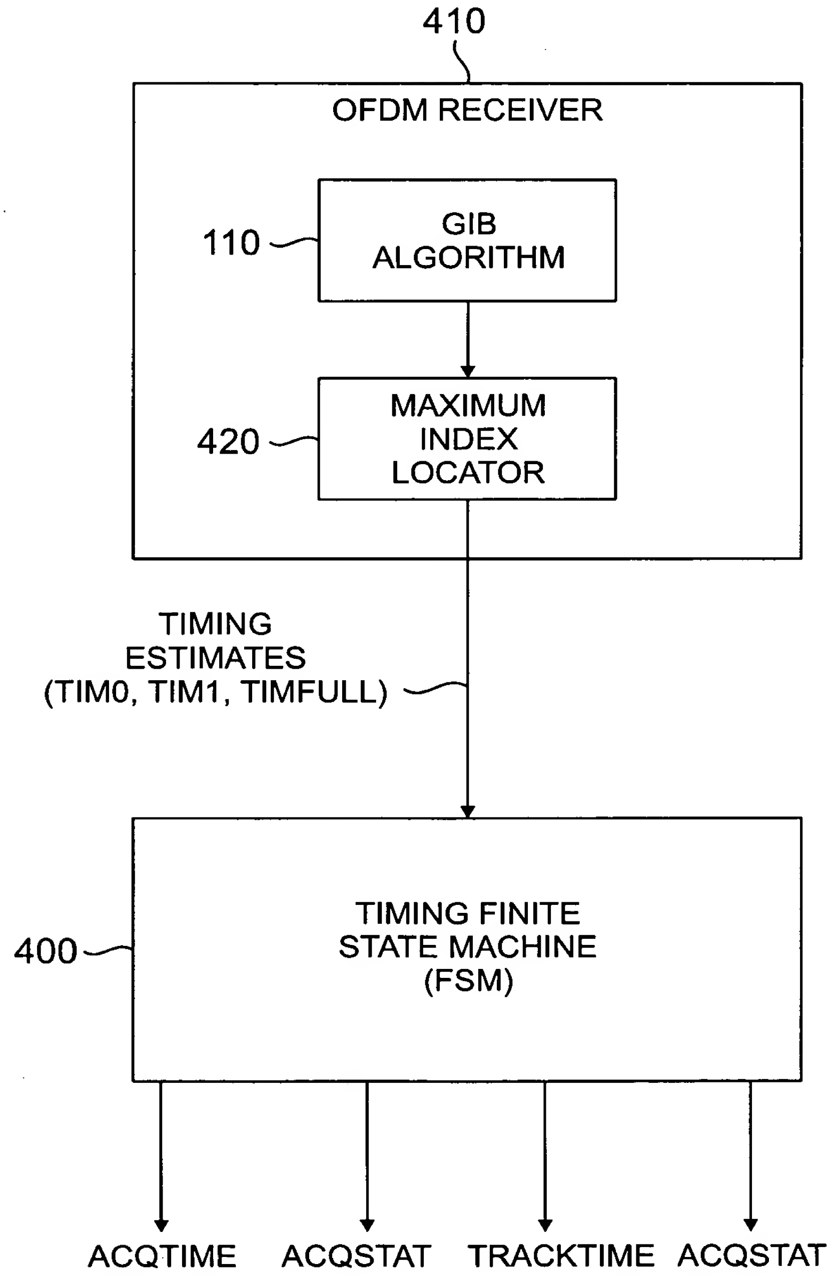

410
OFDM RECEIVER
110
GIB ALGORITHM
420
MAXIMUM INDEX LOCATOR
TIMING ESTIMATES (TIM0, TIM1, TIMFULL)
400
TIMING FINITE STATE MACHINE (FSM)
ACQTIME
ACQSTAT
TRACKTIME
ACQSTAT

FIG. 4

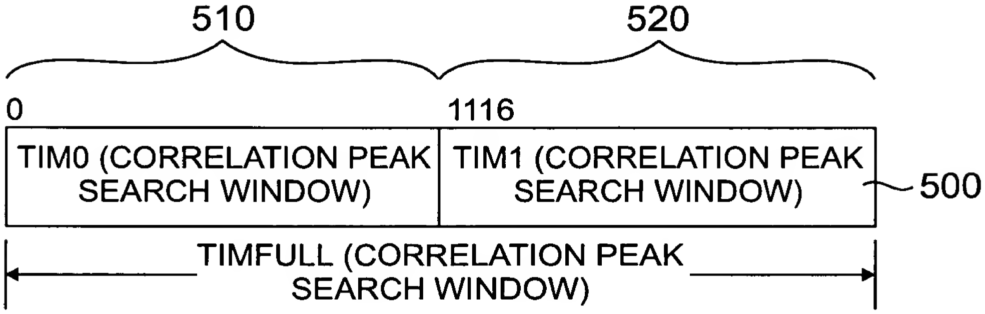
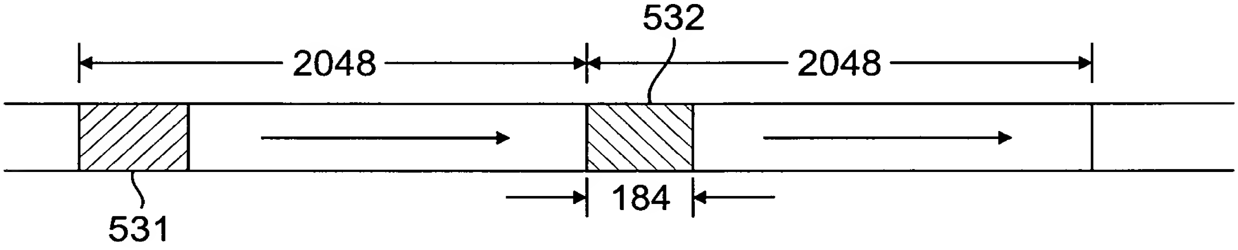

510
520
0
1116
TIM0 (CORRELATION PEAK SEARCH WINDOW)
TIM1 (CORRELATION PEAK SEARCH WINDOW)
500
TIMFULL (CORRELATION PEAK SEARCH WINDOW)

FIG. 5A



CORRELATION OF 184 SAMPLES SPACED 2048 SAMPLES APART. THE PROCESS IS REPEATED TO GET 2232 SAMPLE CORRELATION OUTPUT AS SHOWN BELOW.

FIG. 5B

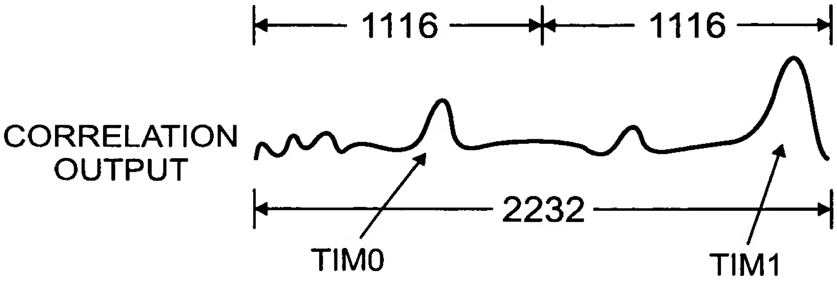


FIG. 5C

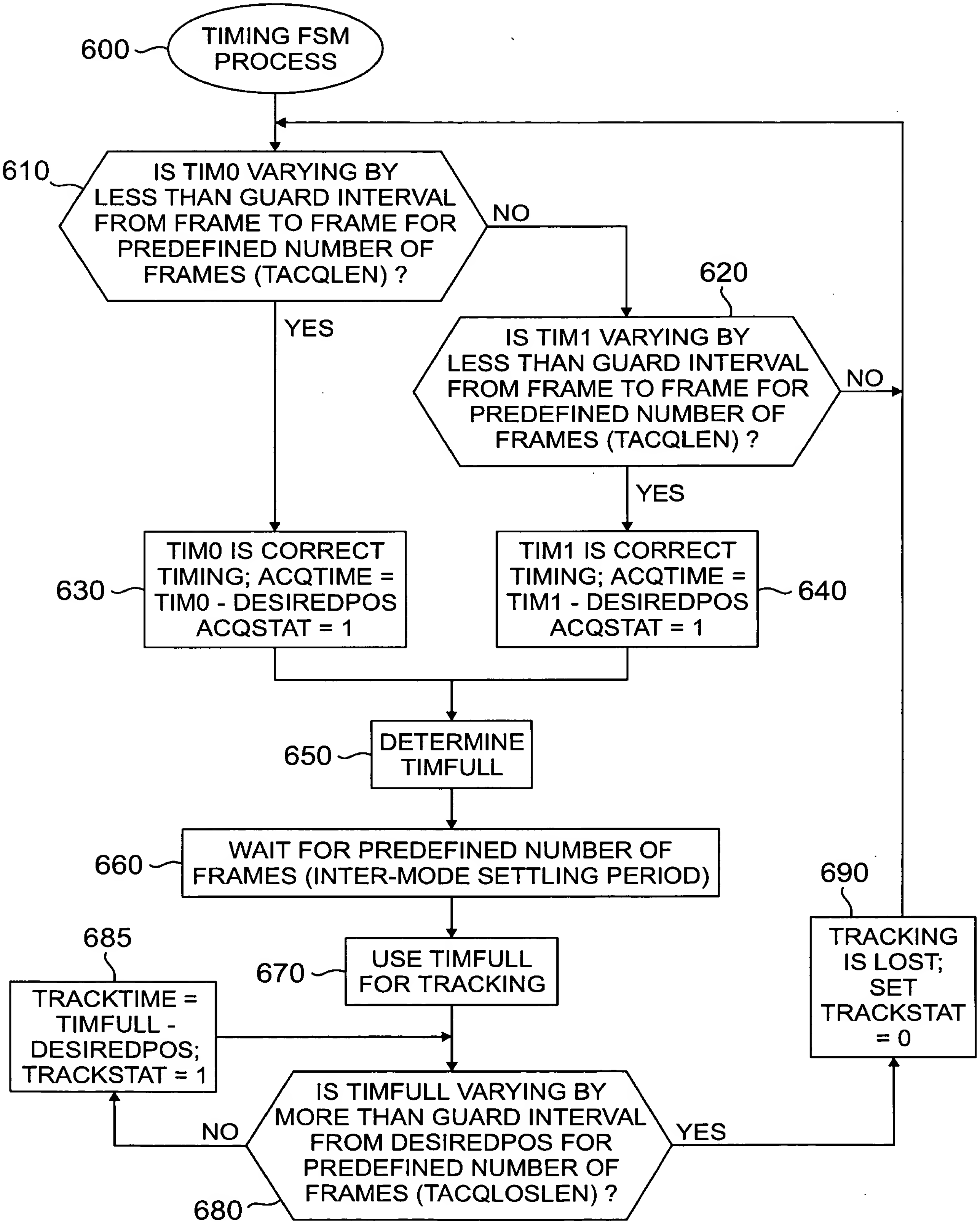

600
TIMING FSM PROCESS
610
IS TIM0 VARYING BY LESS THAN GUARD INTERVAL FROM FRAME TO FRAME FOR PREDEFINED NUMBER OF FRAMES (TACQLEN) ?
NO
YES
620
IS TIM1 VARYING BY LESS THAN GUARD INTERVAL FROM FRAME TO FRAME FOR PREDEFINED NUMBER OF FRAMES (TACQLEN) ?
NO
YES
630
TIM0 IS CORRECT TIMING; ACQTIME = TIM0 - DESIREDPOS ACQSTAT = 1
640
TIM1 IS CORRECT TIMING; ACQTIME = TIM1 - DESIREDPOS ACQSTAT = 1
650
DETERMINE TIMFULL
660
WAIT FOR PREDEFINED NUMBER OF FRAMES (INTER-MODE SETTLING PERIOD)
685
TRACKTIME = TIMFULL - DESIREDPOS; TRACKSTAT = 1
670
USE TIMFULL FOR TRACKING
690
TRACKING IS LOST; SET TRACKSTAT = 0
NO
IS TIMFULL VARYING BY MORE THAN GUARD INTERVAL FROM DESIREDPOS FOR PREDEFINED NUMBER OF FRAMES (TACQLOSLEN) ?
YES
680

FIG. 6

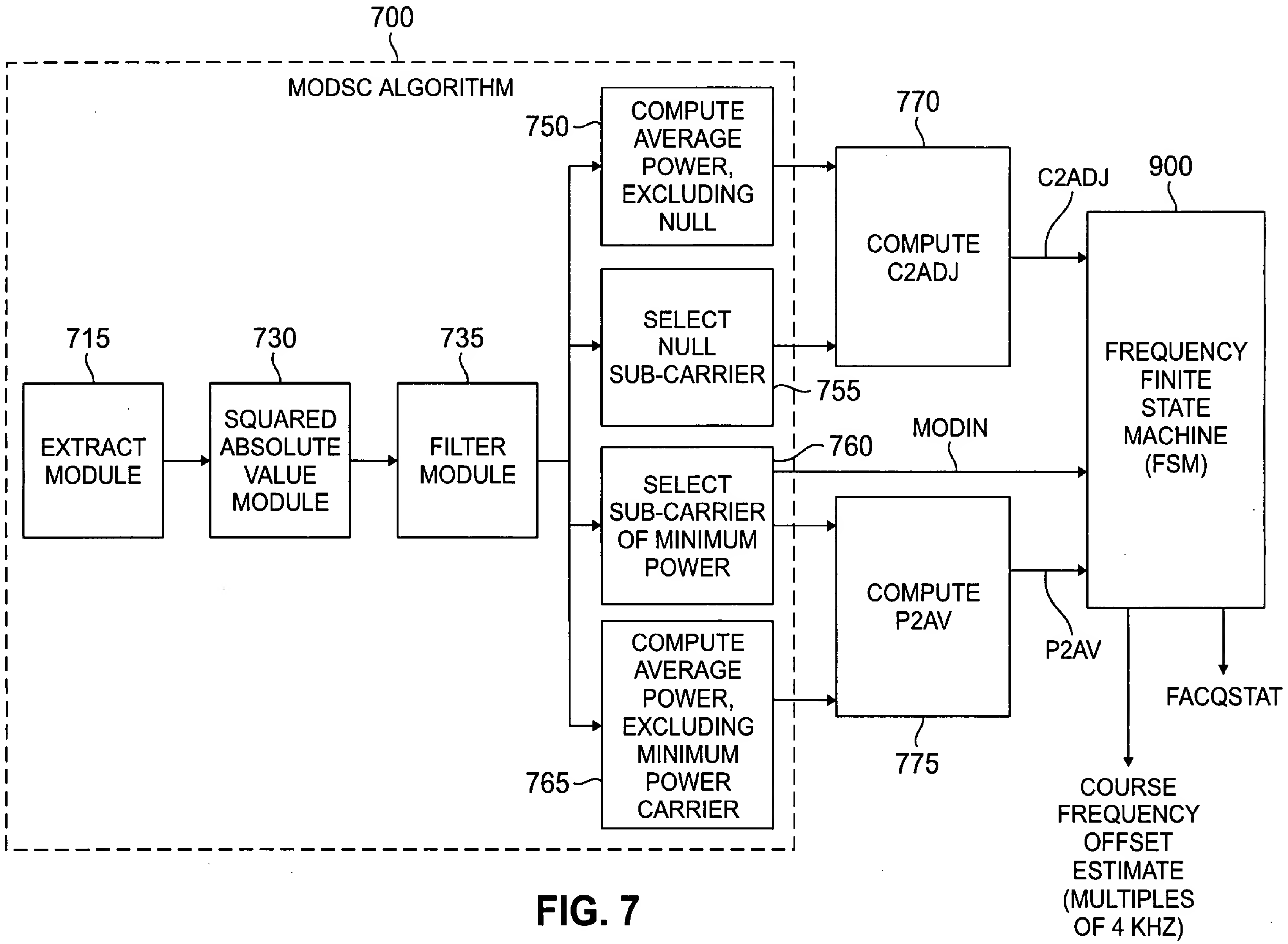

700
MODSC ALGORITHM
715
EXTRACT MODULE
730
SQUARED ABSOLUTE VALUE MODULE
735
FILTER MODULE
750
COMPUTE AVERAGE POWER, EXCLUDING NULL
755
SELECT NULL SUB-CARRIER
760
SELECT SUB-CARRIER OF MINIMUM POWER
765
COMPUTE AVERAGE POWER, EXCLUDING MINIMUM POWER CARRIER
770
COMPUTE C2ADJ
775
COMPUTE P2AV
C2ADJ
MODIN
P2AV
900
FREQUENCY FINITE STATE MACHINE (FSM)
FACQSTAT
COURSE FREQUENCY OFFSET ESTIMATE (MULTIPLES OF 4 KHZ)

FIG. 7

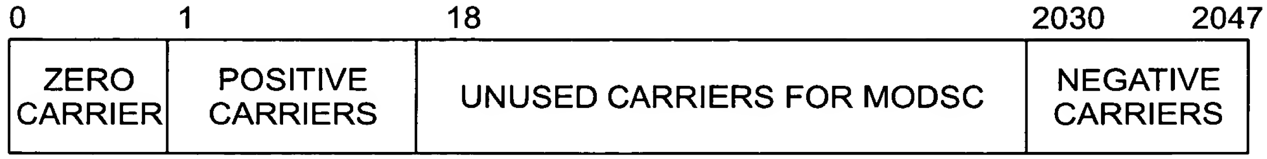

0
1
18
2030
2047
ZERO CARRIER
POSITIVE CARRIERS
UNUSED CARRIERS FOR MODSC
NEGATIVE CARRIERS

FIG. 8

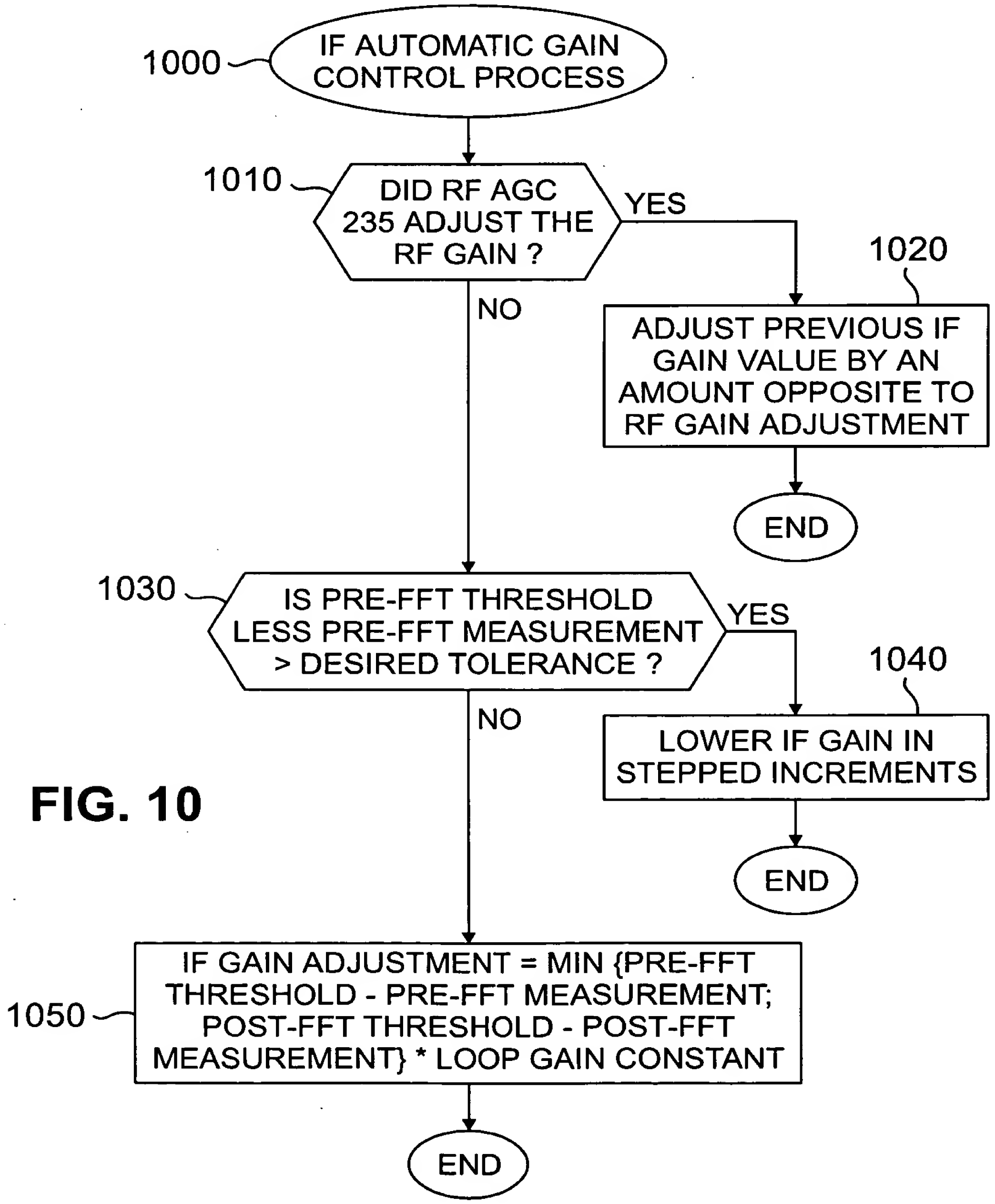

1000
IF AUTOMATIC GAIN CONTROL PROCESS
1010
DID RF AGC 235 ADJUST THE RF GAIN ?
YES
NO
1020
ADJUST PREVIOUS IF GAIN VALUE BY AN AMOUNT OPPOSITE TO RF GAIN ADJUSTMENT
END
1030
IS PRE-FFT THRESHOLD LESS PRE-FFT MEASUREMENT > DESIRED TOLERANCE ?
YES
NO
1040
LOWER IF GAIN IN STEPPED INCREMENTS
END
1050
IF GAIN ADJUSTMENT = MIN {PRE-FFT THRESHOLD - PRE-FFT MEASUREMENT; POST-FFT THRESHOLD - POST-FFT MEASUREMENT} * LOOP GAIN CONSTANT
END

FIG. 10

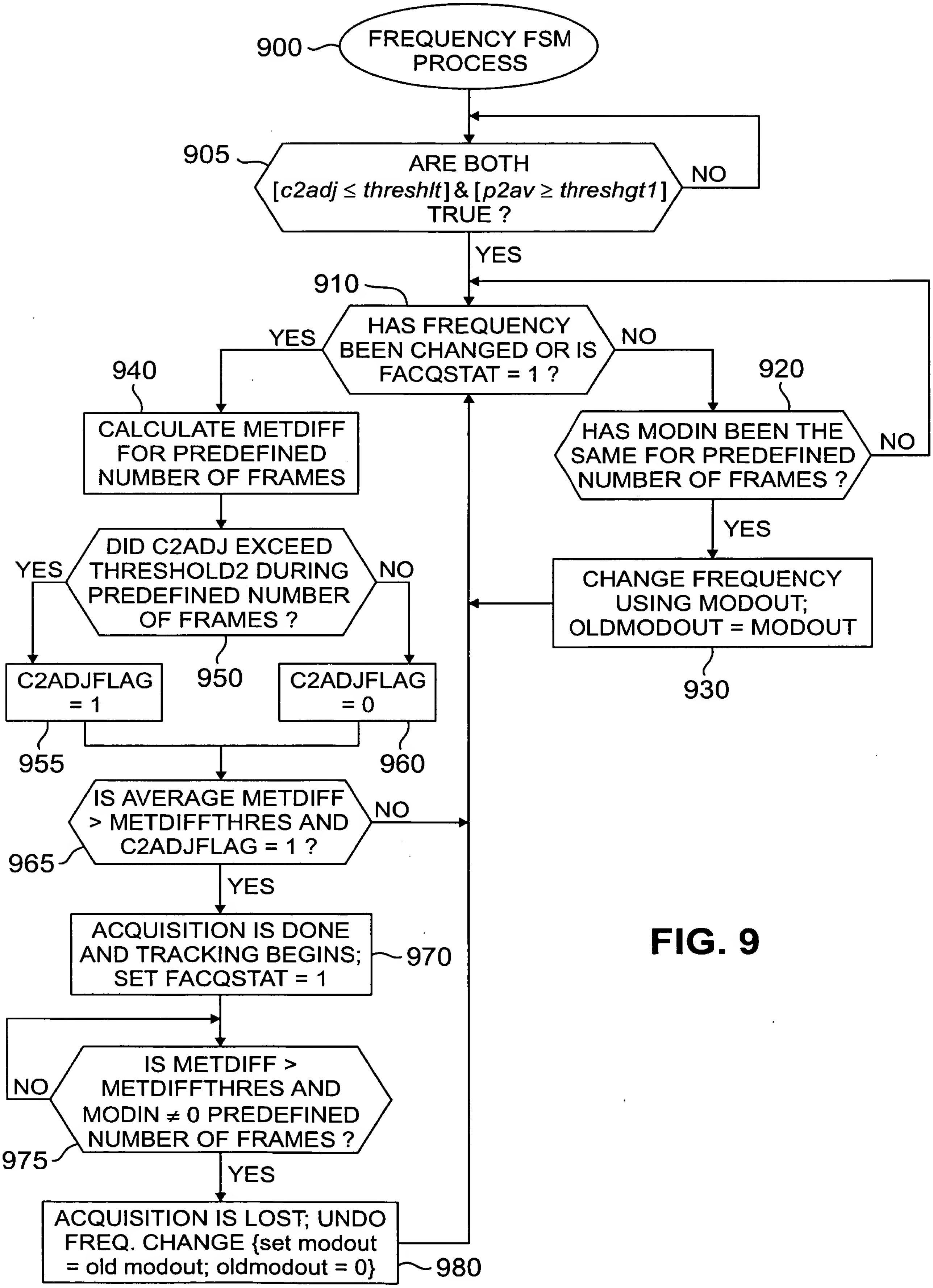

900
FREQUENCY FSM PROCESS
905
ARE BOTH [c2adj ≤ threshlt] & [p2av ≥ threshgt1] TRUE ?
NO
YES
910
HAS FREQUENCY BEEN CHANGED OR IS FACQSTAT = 1 ?
YES
NO
940
CALCULATE METDIFF FOR PREDEFINED NUMBER OF FRAMES
920
HAS MODIN BEEN THE SAME FOR PREDEFINED NUMBER OF FRAMES ?
NO
YES
CHANGE FREQUENCY USING MODOUT; OLDMODOUT = MODOUT
930
DID C2ADJ EXCEED THRESHOLD2 DURING PREDEFINED NUMBER OF FRAMES ?
YES
NO
950
C2ADJFLAG = 1
C2ADJFLAG = 0
955
960
IS AVERAGE METDIFF > METDIFFTHRES AND C2ADJFLAG = 1 ?
NO
965
YES
ACQUISITION IS DONE AND TRACKING BEGINS; SET FACQSTAT = 1
970
IS METDIFF > METDIFFTHRES AND MODIN ≠ 0 PREDEFINED NUMBER OF FRAMES ?
NO
975
YES
ACQUISITION IS LOST; UNDO FREQ. CHANGE {set modout = old modout; oldmodout = 0}
980

FIG. 9

```
INPUT_PORT(1) register float *Prepower;
INPUT_PORT(2) register float *Postpower;
INPUT_PORT(3) register float *RFgain;
OUTPUT_PORT(1) register float *Output; /*IF AGC Gain in dB*/

BLOCKFACTOR long BlockFactor;

PARAMETER(1) float OutputIntervalWidth;/*71 dB*/
PARAMETER(2) float SetPointdBPre;       /*42.2*/
PARAMETER(3) float SetPointdBPost;      /*32.2*/
PARAMETER(4) float Kagc;           /*0.25*/
PARAMETER(5) float PreDropdB;           /*3.0*/
PARAMETER(6) long WaitTime;            /*8 OFDM Frames!!*/

STATE float oldoutput;
STATE float oldrfgain;
STATE long counter;

#include <math.h>

void init ofdmagccontrol2()
{
/*initialize Sum*/
oldoutput = 0.0;
counter = WaitTime;
}

void ofdmagccontrol2()
{
register float dbinpre, dbinpost, err, rfgain, output;
float HalfInterval = (OutputIntervalWidth / 2.0);
```

FIG. 11A

```
LOOP(BlockFactor)

          printf("--------------------IFbeg-----------------\n");
          dbinpre = *Prepower++; dbinpost = *Postpower++;
          rfgain = *RFgain++;

          printf("prepower = %f, post = %f, rfgain = %f\n", dbinpre, dbinpost, rfgain);
          if((rfgain - oldrfgain)! = 0.0)
                    {
                    output = oldoutput - (rfgain - oldrfgain);
                    printf("ifgain = -rfdiff = %f, oldrfgain = %f\n", output, oldrfgain);
                    }
          else if ((SetPointdBPre-PreDropdB - dbinpre <= 0.0) && (counter >= WaitTime))
                    {
                    output = oldoutput - (PreDropdB + 2.0);
                    printf("ifgain = due to Pre = %f\n", - PreDropdB);
                    counter = 0;
                    }
          else
                    {
                    counter++;
                    if(SetPointdBPre - dbinpre < SetPointdBPost - dbinpost)
                              err = SetPointdBPre - dbinpre;
                    else
                              err = SetPointdBPost - dbinpost;
                    err = Kagc*err;
                    output = oldoutput + err;
                    printf("output = %f\n", output);
                    }

          if(output >= HalfInterval)
                    output = HalfInterval;
          else if (output <= -HalfInterval)
                    output = -HalfInterval;
          else
                    output = output;

          *Output++ = output;
          oldrfgain = rfgain;
          oldoutput = output;
          printf("----------------------IFend-----------------------\n");
ENDLOOP
}
```

FIG. 11B